به نام خدا

# فاز چهارم پروژه کامپایلرها و زبان های برنامه‌نویسی

## پاییز ۹۸

مهلت تحویل: ۱۷ دی

---

در این فاز بخش های مربوط به تولید کد را به کامپایلر خود اضافه می‌کنید. در انتهای این فاز، کامپایلر شما به طور کامل پیاده سازی شده و برنامه های نوشته شده به زبان ACTon را به کد قابل اجرا توسط ماشین تبدیل می‌کند. پیاده سازی شما باید به ازای هر فایل ورودی به زبان ACTon، بایت کد معادل آن را تولید کند.
در تست های این فاز، صرفا قابلیت تولید کد کامپایلرتان سنجیده می‌شود و ورودی ها دارای خطاهای نحوی و معنایی که در فازهای قبل بررسی کردید نیستند. اما توجه کنید که شما برای تولید کد به اطلاعات جمع آوری شده در جدول علائم و اطلاعات مربوط به تایپ نود های درخت AST نیاز دارید.

برای سادگی کارتان در این فاز وراثت از مجموعه ی ویژگی های زبان حذف شده است و نیازی به پیاده سازی آن نیست.

### اسمبلر

جهت تولید فایل های .class نهایی از شما انتظار نمی‌رود که فایل باینری را مستقیما تولید کنید. برای این کار می‌توانید از اسمبلر jasmin که در کلاس معرفی شده است استفاده کنید. چنانچه با اسمبلر دیگری قبلا کار کردید و با آن راحت تر هستید می‌توانید از آن استفاده کنید.

### کد مربوط به writeln

جهت نوشتن بر روی صفحه ی نمایش باید از println در کتابخانه ی java.io.PrintStream استفاده کنید. به منظور گرفتن معادل رشته از یک آرایه می‌توان از toString استفاده کرد.

### تساوی اشیاء

گونه های int و boolean با استفاده از مقادیرشان با یکدیگر مقایسه می‌شوند.

برای بررسی == بین انواع دیگر کافی است متد equals را بر روی آن‌ها صدا بزنید.

## عملگرهای && و ||

شما باید این عملیات را به صورت short-circuit پیاده سازی کنید.

### نکات مهم

- یک فایل با فرمت stdID1_stdID2.zip (اگر گروهی تک نفره هستید: stID.zip) آپلود کنید که در آن فایل های شما قرار دارند.
- بایت کد تولید شده به ازای هر کلاس در فایلی با نام classname.j قرار دارد و تمام این فایل ها را در یک فولدر با نام output در کنار فایل های پروژه‌تان ذخیره کنید.
- کد شما باید توسط یک فایل اصلی با نام Acton.java که در کنار کدهای شما وجود دارد، قابل اجرا باشد.

"موفق باشید"